Herausforderungen bei der Integration von Benutzern in Datenorientierten Prozess-Management-Systemen

Vera Künzle und Manfred Reichert

Institut für Datenbanken und Informationssysteme, Universität Ulm
{vera.kuenzle,manfred.reichert}@uni-ulm.de

Zusammenfassung

Im Projekt PHILharmonic Flows entwickeln wir ein datenorientierten Prozess-Management-System der nächsten Generation. In Vorarbeiten haben wir fünf Herausforderungen diskutiert, die eine generische Komponente zur Unterstützung datengetriebener Prozesse mit einer integrierten Sicht auf Daten und Prozesse erfüllen sollte. In diesem Aufsatz betrachten wir zusätzlich die Integration von Benutzern. Dazu stellen wir vier weitere Herausforderungen für die Zugriffskontrolle in datenorientierten Prozess-Management-Systemen vor. Letztgenannte stellen obligatorische und optionale Aktivitäten zur Verfügung. Obligatorische Aktivitäten müssen für den Fortschritt einer Prozessinstanz zwingend ausgeführt werden, optionale Aktivitäten ermöglichen dagegen die Pflege und Verwaltung von Daten unabhängig von der Ausführung eines bestimmten Prozesses. Die Bearbeiterzuordnung für obligatorische Aktivitäten ist dabei nicht nur von der Aktivität an sich abhängig, sondern auch von den Berechtigungen eines Benutzers zur Durchführung der innerhalb der Aktivität erforderlichen Datenänderungen. Berechtigungen für Datenänderungen müssen dazu für verschiedene Objektinstanzen eines Objekttyps jeweils unterschiedlich vergeben werden können. Gleichzeitig darf bei der Ausführung optionaler Aktivitäten die Durchführung von Prozessinstanzen nicht fehlerhaft beeinflusst werden. Weiter erweist sich eine getrennte Verwaltung von Anwendungsdaten und Organisationsmodell als zu unflexibel für eine feingranulare Vergabe von Rechten mit möglichst geringem Administrationsaufwand. Insgesamt bieten datenorientierte Prozess-Management-Systeme eine integrierte Sicht auf Prozesse, Daten und Benutzer, und eröffnen daher völlig neue Anwendungsfelder für Prozess-Management-Technologie.

1 Motivation

Prozess-Management-Technologie bietet vielversprechende Perspektiven für die Computerunterstützung von Geschäftsprozessen unabhängig von einer spezifischen Anwendung. Trotzdem existieren auf dem Software-Markt immer noch spezialisierte Software-Anwendungen (z.B. ERP-, CRM- oder SCM-Systeme). In diesen datenorientierten Systemen ist die Prozesslogik jedoch fest im Anwendungscode "verdrahtet". Ursache dafür ist, dass die derzeit auf dem Markt verfügbaren Workflow-Management-Systeme (WfMS), wie Staffware, AristaFlow BPM Suite [DaRe09a, DaRe09b] und WebSphere ProcessServer, die technologische Reife für die Realisierung der Prozesse in datenorientierten Anwendungen nicht voll erreicht haben. In [KuRe09b] haben wir die Herausforderungen an ein neues Workflow-Paradigma zur Unterstützung von datenorientierten Prozessen vorgestellt. Insbesondere haben wir Anforderungen an eine generische Komponente definiert, welche die von Anwendungssystemen her bekannte daten- und funktionsbezogene Sicht mit einer Sicht auf die jeweils unterstützten Prozesse integriert. In [KuRe09, KuRe09b] sind wir speziell auf die Integration von Prozessen und Daten sowie die Unterstützung datenorientierter Prozesse eingegangen. In diesem Aufsatz diskutieren wir einen komplementären Aspekt – die Integration der Benutzer in datenorientierten Prozess-Management-Systemen. Insbesondere zeigen wir, welche zusätzlichen Herausforderungen hinsichtlich Zugriffskontrolle für Systeme mit integrierter Sicht auf Prozesse und Daten zu erfüllen sind.

Zunächst stellen wir in Kapitel 2 ein einfaches Beispiel vor, entlang dessen wir die nachfolgenden Ausführungen illustrieren. In Kapitel 3 beschreiben wir zunächst den Aufbau und die Arbeitsweise von daten- und funktionsbezogenen Anwendungen sowie von konventionellen WfMS. Im Anschluss fassen wir die Charakteristika datenorientierter Prozess-Management-Systeme, die wir in [KuRe09, KuRe09b] bereits vorgestellt haben, nochmals zusammen. Kapitel 4 gibt einen systematischen Überblick über

die Ziele und Maßnahmen für sichere Informationssysteme im Allgemeinen und über Strategien, Modelle und Mechanismen zur Zugriffskontrolle im Speziellen. In diese Systematik ordnen wir die Anforderungen für die Zugriffskontrolle in datenorientierten Prozess-Management-Systemen ein und grenzen die in diesem Aufsatz betrachteten Herausforderungen von weiterführenden Anforderungen ab. In Kapitel 5 beschreiben wir die Herausforderungen an die Zugriffskontrolle für ein datenorientiertes Prozess Management System im Detail. Kapitel 6 stellt existierende Ansätze vor, die bereits auf einzelne Teil-Problematiken eingegangen sind. Abschließend geben wir in Kapitel 7 einen Ausblick auf unsere zukünftigen Forschungsarbeiten zu datenorientierten Prozess-Management-Systemen.

2 Beispiel

Um unsere nachfolgenden Ausführungen zu illustrieren, stellen wir zunächst als Fallbeispiel den Bearbeitungsprozess einer Bewerbung aus dem Bereich *Human Ressource Management* vor. Der Einfachheit halber gehen wir nur auf Grundfunktionen und allgemeine Abläufe ein:

Anhand eines Online-Formulars im Internet haben Interessenten die Möglichkeit, sich auf eine offene Ausschreibung zu bewerben. Ziel des Prozesses ist es, eine Entscheidung darüber zu treffen, welcher Bewerber zur Besetzung der offenen Stelle eingestellt werden soll. Dazu werden zunächst die Kenntnisse der Bewerber mit den Anforderungen der Stelle verglichen. Verschiedene Sachbearbeiter in der Personalabteilung sind hierbei jeweils für die Bewerbungen zu unterschiedlichen Ausschreibungen zuständig. Zur weiteren Entscheidungsfindung haben die Sachbearbeiter in der Personalabteilung die Möglichkeit, die Bewerbungen zwecks Beurteilung den Führungskräften der jeweils zuständigen Fachabteilungen vorzulegen. Dazu wird für die Mitarbeiter aus den Fachabteilungen von der Personalabteilung jeweils ein Gutachten angelegt. Diese müssen von den betreffenden Mitarbeitern der Fachabteilung ausgefüllt werden. Anhand eines Ausgabedatums kann die Personalabteilung festgelegen, zu welchem Zeitpunkt der jeweilige Mitarbeiter das jeweilige Gutachten erstellen soll. Ist dieses Datum erreicht, kann der Mitarbeiter die zugehörige Bewerbung einsehen, eine Bewertung festlegen und einen Kommentar vermerken. Des Weiteren muss der Mitarbeiter einen Vorschlag für das weitere Verfahren angeben. Dies kann eine Absage für den Bewerber oder die Einladung zu einem Vorstellungsgespräch sein. Nach Rückgabe der Gutachten werden diese von der Personalabteilung ausgewertet und als Entscheidungsgrundlage (bzw. für die Auswahl einer Folgeaktion) verwendet.

3 Grundlagen

3.1 Daten- und funktionsbezogene Anwendungssysteme

Daten- und funktionsbezogene Anwendungen stellen Funktionen zur Verwaltung und Bearbeitung von Daten bereit. Sie basieren dazu typischerweise auf einer (objekt-)relationalen Datenbank. Diese speichert Datenobjekte anhand verschiedener Tabellen (sog. Relationen). Eine Tabelle beschreibt jeweils einen *Objekttyp*. Einzelne Zeilen repräsentieren hierbei die einzelnen *Objektinstanzen*, Spalten definieren die *Attribute* eines Objekttyps bzw. enthalten die *Attributwerte* einer Objektinstanz. Anhand von *Primärschlüsselattributen* können Objektinstanzen eindeutig identifiziert werden. Beziehungen werden anhand von *Fremdschlüsselattributen* beschrieben. Diese Attribute speichern die Werte von Primärschlüsseln anderer Objektinstanzen.

Den Zugang zum System bilden meist Übersichtstabellen, in welchen die einzelnen Objektinstanzen eines bestimmten Objekttyps aufgelistet werden. Ausgehend von diesen Übersichtslisten können instanzspezifisch die *Funktionen zur Einsicht und Änderung der Attributwerte* einer Objektinstanz ausgeführt werden. Die einzelnen `Bewerbungen` werden beispielsweise innerhalb einer spezifischen Übersicht aufgelistet. Für jede Objektinstanz, d.h. für jede `Bewerbung`, kann zum Beispiel eine Funktion zur Änderung der Attributwerte aufgerufen werden. Auch *komplexere Geschäftsfunktionen*, deren Ausführung sich jeweils auf eine bestimmte Objektinstanz bezieht, können an dieser Stelle aufgerufen werden. Eine komplexe Geschäftsfunktion ist z.B. der Vergleich der Kenntnisse eines Bewerbers mit den Anforderungen einer offenen Stelle. Funktionen zur *Neuanlage von Objektinstanzen* sowie komplexere Geschäftsfunktionen, deren Ausführung mehrere Objektinstanzen (auch von verschiedenen Objekttypen) betrifft, können z.B. innerhalb von Programmmenüs aufgerufen werden. Diese Funktionen können keiner spezifischen, bereits existierenden Objektinstanz zugeordnet werden.

Zu erwähnen bleibt, dass die Benutzer dieser Anwendungssysteme meist selbst Teil der Datenbasis sind und als einzelne Objektinstanzen in speziellen Relationen gespeichert werden. Beispiele hierzu sind Autoren und Gutachter in Systemen zur Organisation von wissenschaftlichen Konferenzen, Be-

werber und Mitarbeiter in Personalmanagementsystemen oder Kunden und Sachbearbeiter in Systemen zur Verwaltung von Versicherungsverträgen.

3.2 Konventionelle Workflow-Management-Systeme

In Workflow-Management-Systemen (WfMS) werden innerhalb des *Prozessmodells* einzelne *Aktivitäten* definiert und anhand des *Kontrollflusses* zu Prozessen verknüpft. Letzterer legt die Reihenfolge und Ausführungsbedingungen für die Aktivitäten fest. Dazu stehen z.B. Modellierungskonstrukte für sequentielle, alternative und parallele Abläufe sowie für Schleifen zur Verfügung. Einige WfMS bieten noch zusätzlich fortschrittlichere Konstrukte [AHKB03]. Schließlich wird für die spätere Ausführung jede Prozess-Aktivität mit einem Anwendungsdienst verknüpft.

Innerhalb herkömmlicher WfMS muss zwischen internen und externen Anwendungsdaten differenziert werden. *Interne Anwendungsdaten* werden bei der Festlegung des *Datenflusses* zu Prozessaktivitäten modelliert und über Ein- bzw. Ausgabeparameter mit einzelnen Aktivitäten verknüpft. Hierbei können in den meisten WfMS nur atomare Datenelemente definiert werden. Eine Gruppierung der einzelnen Datenelemente zu Objekten sowie die Definition von Beziehungen zwischen Datenelementen sind in den meisten WfMS dagegen nicht möglich. *Externe Anwendungsdaten* werden von den eingebundenen Anwendungen selbst verwaltet. Auf diese Daten hat das WfMS keinen Zugriff. D.h. es kann nicht gesteuert werden, welche Objekttypen, welche Objektinstanzen und welche Attributwerte jeweils beim Aufruf einer Aktivität gelesen oder bearbeitet werden können. Je nach Implementierung der eingebundenen Funktion des Anwendungsdienstes kann jedoch (z.B. durch Übergabe einer Objekt-ID innerhalb des Datenflusses) geringer Einfluss genommen werden.

Benutzer werden innerhalb eines *Organisationsmodells* verwaltet. Interaktiven Aktivitäten, die eine Aktion des Benutzers erfordern, wird zusätzlich ein *Bearbeiterausdruck* (z.B. eine Benutzerrolle) mit Bezug auf das Organisationsmodell zugeordnet. Das Organisationsmodell wird vollständig getrennt von den internen und externen Anwendungsdaten verwaltet.

Zur Laufzeit wird für jede Ausführung eines Prozesses eine eigene *Prozessinstanz* angelegt. Alle aktivierbaren Aktivitäten werden den jeweils zuständigen Bearbeitern innerhalb ihrer *Arbeitsliste* zur Ausführung angeboten. Bei der Aktivierung wird automatisch die mit der Aktivität verknüpfte Anwendung gestartet und die benötigten Daten geladen [ReDa00, AaHe04].

3.3 Datenorientierte Prozess-Management-Systeme

Erste Erfahrungen mit datenorientierten Prozessen haben wir im Corepro-Projekt gesammelt [MRH07, MRH08]. In [KuRe09b] haben wir die Herausforderungen an eine generische Komponente zur Unterstützung von datenorientierten Prozessen mit integrierter Sicht auf Daten und Prozesse beschrieben. Um nachfolgende Ausführungen zur Zugriffskontrolle in datenorientierten Prozess-Management-Systemen einordnen zu können, fassen wir unsere wichtigsten Erkenntnisse an dieser Stelle nochmals zusammen.

Herausforderung 1: Datenintegration

Anwendungsdaten bestehen aus einer variablen Anzahl von *Objektinstanzen* verschiedener *Objekttypen*, die durch eine Menge von *Attributen* definiert sind. Auch *Beziehungen zwischen Objektinstanzen* werden innerhalb der Objekttypen definiert.

Einer Objektinstanz können mehrere andere Objektinstanzen eines anderen oder desselben Objekttyps zugeordnet sein. Alle Daten können innerhalb von *optionalen Aktivitäten* (unabhängig von Prozessen) zu jedem Zeitpunkt eingesehen und bearbeitet werden. Anwendungsdaten müssen daher vollständig integriert werden. Hierbei müssen Daten anhand von Objekttypen und nicht nur auf Basis atomarer Datenelemente verwaltet werden können. Schließlich müssen variable Mengen von Objektinstanzen sowie deren Beziehungen untereinander berücksichtigt werden können.

Herausforderung 2: Granularität von Prozessen und Aktivitäten

Für die einzelnen Objekttypen existieren jeweils spezifische Bearbeitungsprozesse (sog. *Prozesstypen*). Die einzelnen Aktivitäten innerhalb eines *Prozesstyps* bestehen jeweils aus verschiedenen Aktionen zur Einsicht oder Änderung der Attribute des zugehörigen Objekttyps. Die Instanziierung eines Prozesstyps ist unmittelbar mit der Anlage einer zugehörigen Objektinstanz verknüpft.

Bei der Ausführung einer Prozessinstanz für eine spezifische Objektinstanz sind weiterhin Informationen aus anderen Objektinstanzen, die mit der betreffenden Objektinstanz in Beziehung stehen, rele-

vant. Abhängigkeiten dieser Art werden anhand von Sub-Prozess-Beziehungen zwischen verschiedenen Prozesstypen modelliert. Die Zuordnung der Prozesstypen zueinander entspricht somit der Zuordnung der Objekttypen, d.h. es besteht eine Analogie zwischen Prozess- und Datenstruktur. Durch die variable Anzahl von Objektinstanzen eines Typs ergibt sich eine variable Anzahl von Prozessinstanzen. Diese 1:1-Zuordnung zwischen Objekt- und Prozesstyp bzw. zwischen Objekt- und Prozessinstanz wird in Abbildung 1 illustriert. Für die Objekttypen `Bewerbung` und `Gutachten` existiert jeweils ein eigener Prozesstyp. Bei Anlage einer Objektinstanz wird automatisch eine zugehörige Prozessinstanz erzeugt. Eine Prozessinstanz für eine `Bewerbung` enthält zur Laufzeit genau so viele Sub-Prozessinstanzen, wie es Objektinstanzen für `Gutachten` gibt, die der Objektinstanz der jeweiligen `Bewerbung` zugeordnet sind.

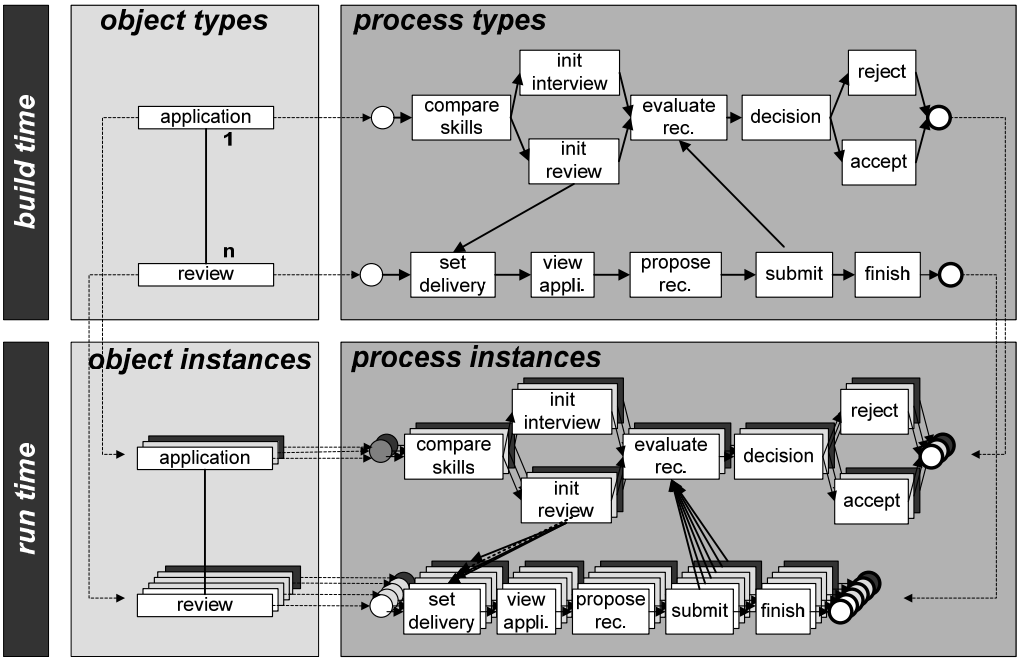

Abbildung 1: Granularität von Prozessen

Herausforderung 3: Datenbasierte Modellierung

Interessanterweise ist der Fortschritt einer Prozessinstanz anhand der Attributwerte der zugehörigen Objektinstanz erkennbar. Folglich sind einzelne *Prozessschritte* hier weniger anhand von Aktivitäten, sondern vielmehr anhand von *Datenbedingungen* definiert. Zur Erreichung eines bestimmten Prozessschritts, d.h. zur Erfüllung einer Datenbedingung, müssen *obligatorische Aktivitäten* ausgeführt werden. Die Aktionen (d.h. die Attributänderungen), die innerhalb einer solchen Aktivität ausgeführt werden müssen, können anhand der Datenbedingungen ermittelt werden. Obligatorische Aktivitäten sind im Gegensatz zu den optionalen Aktivitäten für den Fortschritt einer Prozessinstanz zwingend erforderlich. Abbildung 2 verdeutlicht die datenbasierte Modellierung entlang des Prozesses für ein `Gutachten` einer Bewerbung. Um z.B. ein `Gutachten` an die Personalabteilung zurückzugeben, muss das Attribut `submit` gesetzt werden. Dazu muss dieses Attribut innerhalb des vorhergehenden Prozessschritts geschrieben werden.

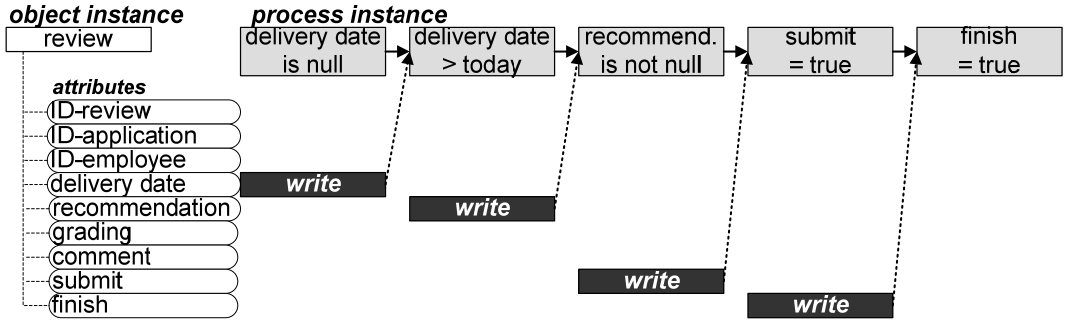

Abbildung 2: Datenbasierte Modellierung

Herausforderung 4: Synchronisation von Prozessen

Eine Prozessstruktur besteht aus hierarchisch angeordneten Prozesstypen. Die Zuordnung von (Sub-) Prozesstypen erfolgt analog zur Datenstruktur. Zur Laufzeit müssen die unterschiedlichen Prozessinstanzen eines Prozesstyps untereinander asynchron ausgeführt werden. Im Gegensatz zu konventionellen WfMS müssen zusätzlich auch die Sub-Prozessinstanzen jeweils asynchron zur übergeordneten Prozessinstanz ausgeführt werden können. Bei dieser Synchronisation müssen die verschiedenen Mengenbeziehungen berücksichtigt werden.

Für jede `Bewerbung` existieren zur Laufzeit eine eigene Objekt- und Prozessinstanz. Die Anzahl der untergeordneten Prozessinstanzen eines Bewerbungsprozesses entspricht der jeweiligen Anzahl an `Gutachten` die der Bewerbung zugeordnet sind. Während der Ausführung der Prozessinstanzen für die `Gutachten` muss die Prozessinstanz für die `Bewerbung` weiter ausgeführt werden können. Innerhalb der Prozessinstanz der `Bewerbung` werden die Ergebnisse der `Gutachten` ausgewertet. Diese Auswertung kann erst erfolgen, wenn alle `Gutachten` zurückgegeben worden sind. Die Anzahl der `Gutachten` wiederum kann von Bewerbung zu Bewerbung unterschiedlich sein, d.h. sie ist zur Laufzeit variabel.

Herausforderung 5: Flexibilität

Welche Aktivitäten zu einem bestimmten Zeitpunkt ausgeführt werden können, ist abhängig vom aktuell erreichten Prozessschritt, d.h. der aktuell gültigen Datenbedingung und nicht von der Ausführung anderer Aktivitäten. Dies gilt sowohl für obligatorische als auch optionale Aktivitäten. Welche Attribute bei Bearbeitung einer Objektinstanz jeweils gelesen oder geschrieben werden können, ist abhängig vom Fortschritt der zur Objektinstanz gehörenden Prozessinstanz. Dadurch können Aktivitäten, solange die jeweils relevante Datenbedingung weiterhin erfüllt ist, auch wiederholt ausgeführt werden. Abbildung 2 verdeutlicht die Abhängigkeit von Lese- und Schreibberechtigungen für Attributwerte einer Objektinstanz eines `Gutachtens` vom Fortschritt der zum `Gutachten` gehörenden Prozessinstanz.

4 Zugriffskontrolle

Informationssysteme stellen Ressourcen in Form von Daten und Funktionen zur Verfügung. Dabei müssen die Interessen von Benutzern bezüglich *Verfügbarkeit, Integrität und Vertraulichkeit* erfüllt werden [Ber98, FK92]. Insbesondere müssen der Zugang zu Daten und die Ausführung von Funktionen vor unberechtigten Zugriffen (*Vertraulichkeit*) und unzulässigen Änderungen (*Integrität*) geschützt werden. Gleichzeitig muss gewährleistet werden, dass jeder Benutzer Zugang zu allen benötigten Daten und Funktionen hat (*Verfügbarkeit*).

Um dies zu erreichen, müssen unterschiedliche Maßnahmen implementiert werden. Hierzu ist zunächst einmal eine geeignete Identifizierung der Benutzer, die *Authentifizierung*, erforderlich. Diese bildet die Basis für die *Zugriffskontrolle*, d.h. die *Autorisierung*, sowie für weitere *Mechanismen* (z.B. *Verschlüsselung*). Auf Grundlage der *Zugriffskontrolle* bzw. *Zugriffsrechte* kann kontrolliert werden, welche *Subjekte* (z.B. Benutzer) in welcher Form auf welche *Objekte* (z.B. Daten, Funktionen oder Aktivitäten) des Systems zugreifen können. Zugriffskontrolle kann auf verschiedenen Abstraktionsebenen betrachtet werden [SaVi01]: *Strategien, Modelle und Mechanismen* (für eine ähnliche Kategorisierung siehe [San00]).

Strategien sind allgemeine Festlegungen, welche Komponenten (z.B. Objekte, Funktionen, Aktivitäten, etc.) innerhalb eines Systems geschützt werden müssen. Sie legen die prinzipielle Vorgehensweise, sowie und die benötigten Zugriffsrechte fest. Innerhalb von *Modellen* wird die formale Repräsentation der Strategie definiert (z.B. HRU-Modell[1] [SaVi01]). Dadurch können Zugriffe vom System kontrolliert und überwacht werden. Die konkrete technische Implementierung wird anhand eines *Zugriffskontrollmechanismus* festgelegt [SaVi01].

Unterschiedliche Systeme stellen unterschiedliche Anforderungen an die jeweils benötigte Strategie zur Zugriffskontrolle. Die Zugriffskontrolle kann dazu, je nach Systemfunktionalität, in vier Stufen eingeteilt werden [Pfe05]. Auf der ersten Stufe befinden sich Anforderungen an die Zugriffskontrolle in Informationssystemen im Allgemeinen, dieser Kategorie sind daten- und funktionsorientierte Anwendungen zuzuordnen. Anforderungen an die Zugriffskontrolle in prozessorientierten Informationssystemen, zu denen auch WfMS zählen, bilden die zweite Stufe. Bietet ein System die Möglichkeit Prozesse zur

[1] Modell nach Harrison, Ruzzo, Ullman (basiert auf 6 Elementaroperationen)

Laufzeit anzupassen [ReRD09], entstehen zusätzliche Anforderungen die innerhalb der dritten Stufe angeordnet sind. Auf der vierten Stufe sind Anforderungen angeordnet, die entstehen, wenn Prozesse oder Funktionalitäten über Bereichs-, Unternehmens- oder Sicherheitsgrenzen hinaus reichen.

Abbildung 3 stellt die unterschiedlichen Abstraktionsebenen für die Zugriffskontrolle sowie die verschiedenen Stufen für unterschiedliche Systeme dar.

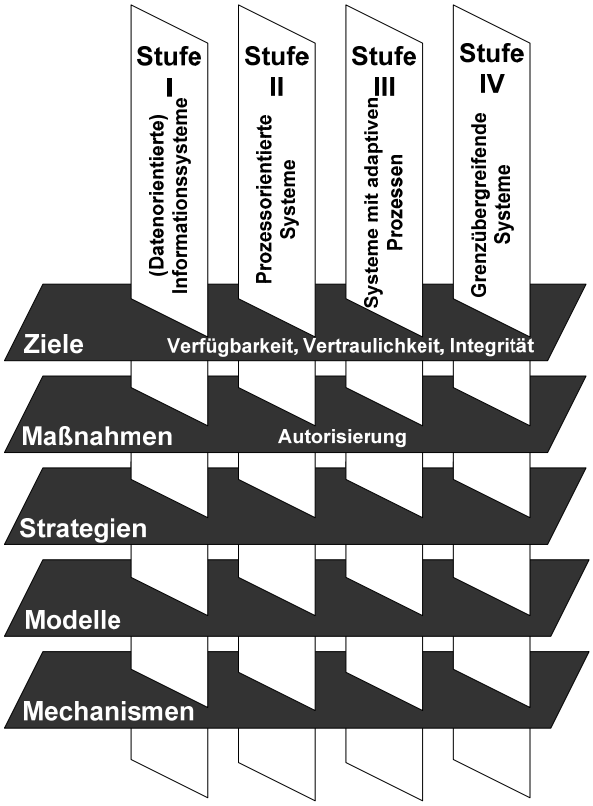

Abbildung 3: Abstraktionsebenen und Stufen der Zugriffskontrolle

Je nach Art des Systems, d.h. abhängig von der betrachteten Stufe, muss die benötigte Strategie unterschiedliche Arten von Rechten unterstützen. Rechte können in *Administrations-, Lese-, Ausführungs-, Eingriffs- und Änderungsrechte* eingeteilt werden [Pfe05]. Abbildung 4 illustriert die für jede Stufe benötigten Arten von Rechten.

	Stufe I	**Stufe II**	**Stufe III**	**Stufe IV**
Ausführungsrechte	**Funktionen**	**Aktivitäten**	wie Stufe II	Integration verschiedener Strategien
		Instanziierung		
Leserechte	**Objekttypen**	(intern verwaltete)	wie Stufe II	
	Objektinstanzen	**Daten**		
	Objektattribute	Prozessmodelle Organisationsmodell Datenmodell Funktionsmodell		
		Prozessinstanzen		
Administrationsrechte	Rechte	Rechte	wie Stufe II	
		Modelle		
Eingriffsrechte	Berechtigungen	Bearbeiterzuordnungen	wie Stufe II	
		Prozessinstanzen		
Änderungsrechte	Rechte		Rechte	
			Prozessinstanzen	
			Prozessmodelle	
			Modelle	

Abbildung 4: Rechtearten pro Systemstufe

Weiterhin können Zugriffsrechte in Bezug auf die *explizite oder implizite Zuordnung* [Pfe05] sowie in Bezug auf eine *passive oder aktive Aktivierung* [Tho97] des Rechts kategorisiert werden. Unter expliziten Rechten versteht man die direkte Zuordnung zu Benutzern oder Rollen, implizite Rechte werden automatisch durch direkte Zuordnung eines anderen Rechts vergeben. Ein Benutzer wird beispiels-

weise explizit als Bearbeiter einer Aktivität zugeordnet und erhält damit implizit die Berechtigungen zu den von der Aktivität zur Verfügung gestellten Anwendungsdaten. Ist zur Modellierzeit entscheidbar, ob einem Benutzer eine entsprechende Berechtigung zugeordnet werden kann, spricht man von einem *passiven Recht*. Bei *aktiven Rechten* kann dagegen erst zur Laufzeit, z.B. abhängig von der Uhrzeit, entschieden werden, ob die Berechtigung vergeben werden kann.

In diesem Aufsatz beschreiben wir die Anforderungen an eine Strategie für die Zugriffskontrolle in datenorientierten Prozess-Management-Systemen. Wir beschränken uns auf *Ausführungsrechte für Funktionen und Aktivitäten* sowie auf *Leserechte für Anwendungsdaten*. Diese Rechtearten sind in Abbildung 4 hervorgehoben. Auf Administrations-, Eingriffs- und Änderungsrechte gehen wir nicht ein. Des Weiteren bleiben Rechte zur Instanziierung von Prozessmodellen sowie Leserechte für unterschiedliche Modelle und Prozessinstanzen zunächst ausgeblendet.

4.1 Grundlegende Strategien: DAC, MAC und RBAC

Zu den grundlegenden Zugriffsstrategien gehören *Discretionary Access Control (DAC)* [Ber98], *Mandatory Access Control (MAC)* [Ber98] sowie *rollenbasierte Zugriffskontrolle (RBAC)* [SaVi01].

Bei der DAC kann festgelegt werden, welche Rechte (z.B. Lese- oder Schreibberechtigungen) ein Benutzer jeweils auf die im System vorhandenen Objekte hat, d.h. die Vergabe von Berechtigungen erfolgt identitätsbezogen pro Benutzer [Ber98, FK92]. Ein Zugriffsrecht kann anhand einer Relation von Subjekt, Objekt und Recht beschrieben werden. Die Vergabe der Berechtigungen erfolgt jeweils durch den Eigentümer des Objekts.

MAC verfolgt einen konträren Ansatz. Die Vergabe von Berechtigungen erfolgt auf Basis von Regeln. Dazu werden Subjekte und Objekte in verschiedene Klassen (z.B. Sicherheitsstufen) eingeteilt. Zwischen diesen Klassen werden Beziehungen für die jeweils erlaubten Zugriffe definiert [Ber98].

In RBAC schließlich werden Berechtigungen nicht benutzerspezifisch, sondern anhand von Rollen vergeben. Benutzern werden zu diesem Zweck entsprechende Rollen zugeordnet [FeKu92, SCFY96]. Rollen bilden eine zusätzliche Abstraktionsebene zwischen Subjekten und Objekten, und erlauben eine weitaus weniger aufwändige Administration. Die Vergabe von Rechten erfolgt durch einen Administrator und nicht durch die Eigentümer der jeweiligen Objekte. Dadurch können einheitlichere Regeln definiert werden. Anhand von rollenbasierten Strategien können sowohl individuelle als auch regelbasierte Strategien umgesetzt werden [OSM00].

Durch Weitergabe von Berechtigungen durch den Eigentümer von Objekten kann bei der Verwendung von DAC meist keine einheitliche Strategie verfolgt werden [Ber98]. In vielen Szenarien sind Benutzer nicht zwingend Eigentümer der Objekte auf die sie Zugriff haben [FeKu92]. Die Vergabe von Rechten an jeden einzelnen Benutzer erfordert einen hohen Administrationsaufwand [Ber98]. MAC ist eher eindimensional auf die Beschränkung von möglichen Informationsflüssen ausgerichtet. Die meisten Anwendungen erfordern jedoch eine Zugriffskontrolle sowohl in Bezug auf Daten als auch in Bezug auf die Funktionen die innerhalb der Anwendung zur Verfügung stehen. Hierbei sollte nicht nur die Funktion an sich, sondern auch die Datenmenge, die als Eingabeparameter für die jeweilige Funktion verwendet werden kann, berücksichtigt werden [FeKu92]. Ein Benutzer sollte zu jedem Zeitpunkt der Systemausführung nur die Rechte besitzen, die er tatsächlich für die Ausführung seiner Aufgaben benötigt. Dieses Konzept wird in vielen Arbeiten "Principle of Least Privilege" [FeKu92] genannt. Bei einer benutzerbasierten Autorisierung, wie sie in DAC und MAC verfolgt wird, ist dies nur schwer sicherzustellen. Des Weiteren kann nicht systemunterstützt gewährleistet werden, dass für Benutzer mit den gleichen Aufgaben auch die gleichen Berechtigungen vergeben werden [FeKu92].

Sowohl in den meisten daten- und funktionsorientierten Anwendungen als auch innerhalb von konventionellen WfMS werden mittlerweile rollenbasierte Strategien verwendet [FeKu92, KS01]. Diese Strategien können einheitlich von einem System-Administrator konfiguriert werden und erlauben die konsistente Vergabe von Rechten in Bezug auf die Funktionen und Aufgaben, die ein Benutzer innerhalb eines Unternehmens besitzt. Die zusätzliche Abstraktionsebene zwischen Benutzern und Berechtigungen erlaubt eine schnellere und weniger aufwendige Administration der Rechte, da die Anzahl an Rollen in einem Unternehmen bedeutend kleiner ist als die Anzahl an Benutzern [BFA99]. Des Weiteren müssen bei Positionswechseln von Benutzer innerhalb eines Unternehmens die Berechtigungen nicht jedes Mal angepasst werden [BFA99, OSM00].

4.2 Zugriffskontrolle in daten- und funktionsbezogenen Anwendungen

Die meisten daten- und funktionsbezogenen Anwendungen verfolgen bei der Zugriffskontrolle eine rollenbasierte Strategie. Das jeweilige Modell sowie der verwendete Mechanismus (d.h. die konkrete Implementierung) sind jedoch von System zu System sehr unterschiedlich und in der Regel anwendungsspezifisch realisiert. Anwendungssysteme für die reine Datenverwaltung regeln beispielsweise lediglich den Zugriff auf Objekte. Ein Benutzer kann dadurch alle vom System angebotenen Funktionen für sämtliche Objekte in seinem Zugriff ausführen. Andere Systeme, wie z.B. webbasierte Anwendungen, beschränken den Zugriff rollenbasiert auf bestimmte Seiten.

Im Folgenden beschreiben wir die Anforderungen, die eine Zugriffskontrolle für daten- und funktionsbezogene Anwendungssysteme (d.h. Systeme der Stufe I [Pfe05]) erfüllen sollte. Wie erwähnt beschränken wir uns in diesem Aufsatz auf Lese- und Ausführungsrechte.

Granularität von Objekten: Die Vergabe von Berechtigungen sollte möglichst feingranular in Bezug auf einzelne Attribute von Objekten möglich sein. Sowohl Mitarbeiter mit der Rolle `Personalabteilung` als auch Mitarbeiter mit der Rolle `Fachabteilung` dürfen Objektinstanzen vom Objekttyp `Gutachten` bearbeiten. Mitarbeiter der `Fachabteilung` dürfen jedoch andere Attribute der Objektinstanz bearbeiten als Mitarbeiter der `Personalabteilung`. Hierbei sollte weiterhin differenziert werden können, ob eine Berechtigung für alle Objektinstanzen eines Objekttyps oder nur für eine einzelne Objektinstanz bzw. eine bestimmte Teilmenge von Objektinstanzen gültig ist [KKC02, LS97, TS97, HW04]. Ein Mitarbeiter mit der Rolle `Fachabteilung` sollte beispielsweise nur `Gutachten` bearbeiten dürfen, die auch von der `Personalabteilung` für ihn angelegt wurden. Eine Vergabe von Zugriffsrechten in Bezug auf einzelne Objektinstanzen ist sehr aufwändig. Problematisch ist zusätzlich, dass die Objektinstanzen bei der Administration der Rechte zur Modellierzeit in der Regel nicht bekannt sind bzw. noch nicht existieren. Des Weiteren ist der Zugriff auf einzelne Objektinstanzen meist nicht abhängig von einer Rolle sondern von einem konkreten Benutzer. Dies geht über die Basisstrategien der rollenbasierten Zugriffskontrolle hinaus [SCFY96]. Anforderungen dieser Art sind innerhalb der meisten Systeme deshalb immer noch hart im Anwendungscode verdrahtet [KKC02].

Kontext: Auch der *Kontext*, in dem ein bestimmtes Recht zur Anwendung kommt, ist relevant [KKC02]. Hiermit sind beispielweise ein Ereignis oder eine Zeitangabe gemeint. Ereignisse spiegeln sich anhand von Attributwerten bestimmter Objektinstanzen wider. Ein Mitarbeiter der `Fachabteilung` darf ein `Gutachten` nur solange bearbeiten, bis er es an die `Personalabteilung` zurückgegeben hat. Zur Rückgabe eines `Gutachtens` setzt der Mitarbeiter einen entsprechenden Wert innerhalb des Attributs `submit` des Gutachtens. Zeitangaben können absolute Angaben oder relative Angaben (z.B. im Bezug auf die Ausführung einer anderen Funktion) sein [HuWe04]. Die Rückgabe eines `Gutachtens` muss beispielsweise innerhalb von 14 Tagen erfolgen.

Aufgabentrennung und Aufgabenbindung: Bestimmte Rollen innerhalb eines Unternehmens schließen sich gegenseitig aus ("Separation of Duties") [SCFY96]. Des Weiteren können sich verschiedene Funktionen in Bezug auf ein Objekt gegenseitig ausschließen. Auch Mitarbeiter können sich auf ausgeschriebene Stellen bewerben. Ein Mitarbeiter sollte jedoch kein `Gutachten` für seine eigene `Bewerbung` erstellen können. Um Betrugsversuche zu vermeiden, müssen diese Funktionen von jeweils unterschiedlichen Benutzern durchgeführt werden (*Aufgabentrennung*). In anderen Fällen erfordern verschiedene Funktionen die Ausführung von jeweils demselben Benutzer (*Aufgabenbindung*). Hierzu müssen Bedingungen in Bezug auf Benutzer, Rollen und Rechte formuliert werden können [SCFY96].

4.3 Zugriffskontrolle in konventionellen WfMS

Ausführungsrechte werden innerhalb von WfMS in Form von Bearbeiterausdrücken für Aktivitäten vergeben [KaSa01, RiRe08]. Für jede Aktivität, die eine Interaktion des Benutzers erwartet, kann ein Bearbeiterausdruck in Bezug auf das *Organisationsmodell* definiert (z.B. eine Rolle) werden [RHE04]. Kann eine Aktivität zur Laufzeit aktiviert werden, wird sie den jeweiligen Bearbeitern innerhalb von Arbeitslisten zugeordnet. Verschiedene Arten und Möglichkeiten für die genaue Zuordnung zur Laufzeit werden in [RHE04] (*Workflow Resource Patterns*) beschrieben.

Anwendungsdaten können in WfMS intern und extern verwaltet werden. Externe Anwendungsdaten werden außerhalb des WfMS von den jeweils eingebundenen Anwendungsdiensten bzw. -systemen verwaltet und können vom WfMS nur sehr eingeschränkt beeinflusst werden. Je nach Implementierung der Aktivitäten kann anhand von Übergabeparametern (z.B. ID einer Objektinstanz) die Daten-

auswahl innerhalb einer Aktivität beeinflusst werden. Zugriffrechte auf Anwendungsdaten, die innerhalb des WfMS verwaltet werden, werden implizit bei der Ausführung von Aktivitäten vergeben. Diese Leserechte gelten jeweils nur für die Dauer der Ausführung der entsprechenden Aktivität [Kin97].

Aufgabentrennung und Aufgabenbindung: Bedingungen zur Aufgabentrennung und -bindung wie sie Systeme der Stufe I [Pfe05] erfordern, sind gerade für Aktivitäten eines Prozesses von großer Bedeutung [KaSa01, RHE04].

Granularität von Objekten: In den meisten WfMS können lediglich atomare Datenelemente verwaltet werden, die Gruppierung zu Objekten ist nicht möglich. Es gibt jedoch fortschrittlichere Ansätze, die auch auf die Vergabe von Datenberechtigungen anhand von Objekten und deren Attribute [Bot02] eingehen, sowie auf den Zugriff auf einzelne Objektinstanzen [SSML02] erlauben.

Kontext: Innerhalb von prozessorientierten Anwendungssystemen ist in erster Linie der Ausführungskontext, in welchem ein Recht verwendet wird, von Bedeutung. Rechte für Datenzugriffe sollten für jede Aktivität spezifisch vergeben werden können [Bot02, T97].

4.4 Zugriffskontrolle in adaptiven WfMS

Adaptive WfMS [Rei00, WSR09, ReRD09, RiRe06] bieten die Möglichkeit, Prozesse zur Laufzeit anzupassen und sind daher der dritten Stufe zugeordnet (siehe Abbildung 3). Wie Abbildung 4 zeigt, müssen in Systemen dieser Art zusätzlich Änderungsrechte berücksichtigt werden. Für aktivitätsorientierte WfMS haben wir innerhalb des ADEPT-Projekts bereits einige Aspekte in diesem Bereich untersucht. [WRWR05] etwa beschreibt einen Ansatz zur Definition von Änderungsrechten an Prozessmodellen auf Schemaebene und zur Laufzeit. Hierbei kann bei der Vergabe von Berechtigungen für Prozessänderungen der jeweilige Prozess-Kontext berücksichtigt werden. Der Ansatz ermöglicht die Einschränkung der Änderungsoperationen auf bestimmte Benutzer, ohne dabei die durch diese Systeme gewonnene Flexibilität unnötig einzuschränken.

Zusätzlich zu den Änderungen an Prozessmodellen und Prozessinstanzen müssen auch Änderungen an Organisationsmodellen sowie Änderungen bei der Vergabe der Rechte selbst (z.B. den Bearbeiterzuordnungen) berücksichtigt werden (siehe Abbildung 4). Diese Aspekte haben wir innerhalb des Projekts CEOSIS [LRDR06, RiRe05, RiRe07, RiRe08] untersucht. Hierbei wurden innerhalb eines Rahmenwerks, semantisch eindeutige Operatoren für Änderungen an Organisationsmodellen definiert [RiRe07]. Da sich Bearbeiterzuordnungen auf die Komponenten des Organisationsmodells beziehen, können Änderungen am Organisationsmodell selbst zu Inkonsistenzen in Bezug auf die definierten Bearbeiterausdrücke führen. Um dies zu vermeiden sind in [RiRe07] zusätzlich Lösungen beschrieben, die die (teil-automatische) Anpassung der Bearbeiterausdrücke bei Anwendung der definierten Änderungsoperatoren unterstützen. Darüber hinaus wurden innerhalb des CEOSIS-Projekts auch mögliche Seiteneffekte solcher Änderungen analysiert. Änderungen an Formeln für die Bearbeiterzuordnungen wirken sich auf die Menge der möglichen Bearbeiter aus und erfordern daher eine Anpassung der Arbeitslisten [RiRe08, RiRe09].

4.5 Zugriffskontrolle in grenzübergreifenden WfMS

Auf der vierten Stufe sind Anforderungen angeordnet, die entstehen, wenn Prozesse oder Funktionalitäten über Bereichs-, Unternehmens- oder Sicherheitsgrenzen hinaus reichen (siehe Abbildung 3). Hierbei entstehen neue Herausforderungen da heterogene Zugriffskontrollstrategien vereinheitlicht werden müssen (siehe Abbildung 4). Innerhalb des Projekts PROVIADO [BBR06] haben wir Ansätze für ein flexibles Monitoring von Prozessen in verteilten Umgebungen beschrieben. Hierbei wurden auch Lösungen für Zugriffkontrollstrategien in Bezug auf das Monitoring solcher Prozesse berücksichtigt. [BRBB09] etwa beschreibt einen Ansatz zur Definition einer neuen Zugriffskontrollstrategie in welcher die Strategien der involvierten Systeme und Prozesse berücksichtigt werden.

5 Anforderungen an die Zugriffskontrolle für datenorientierte Prozess-Management-Systeme

Ziel von datenorientierten Prozess-Management-Systemen ist es, Benutzern eine integrierte Sicht auf Daten, Funktionen und Prozesse zur Verfügung zu stellen [KuRe09, KuRe09b]. Dies bringt auch neue Herausforderungen an die Integration von Benutzern mit sich [KuRe09c]. Die Strategien für die Zugriffskontrolle in daten- und funktionsorientierten Anwendungen d.h. Anwendungssystemen der Stufe I [Pfe05], müssen mit den Strategien für die Zugriffskontrolle in WfMS (d.h. prozessorientierte Anwendungssysteme der Stufe II [Pfe05]) kombiniert und integriert werden.

In diesem Kapitel beschreiben wir die zusätzlichen Herausforderungen an die Zugriffskontrolle bei Systemen mit integrierter Sicht auf Daten und Prozesse. Bisherige Anforderungen an die Zugriffskontrolle für daten- und funktionsorientierte Anwendungen sowie für WfMS bleiben auch für datenorientierte Prozess-Management-Systeme relevant. Dazu zählen:

- Bearbeiterzuordnungen
- Zugriffe auf Daten, d.h. Zugriffe auf bestimmte Objekttypen und deren Attribute sowie Zugriffe auf individuelle Objektinstanzen
- Berücksichtigung des Kontexts
- Unterstützung von Konzepten zur Aufgabentrennung und -bindung

Datenorientierte Prozess-Management-Systeme stellen Basisfunktionen zur Einsicht und Bearbeitung von Objektinstanzen verschiedener Objekttypen sowie zur Anlage von neuen Objektinstanzen zur Verfügung. Weiterhin können komplexere Funktionen sowohl für einzelne Objektinstanzen als auch in Bezug auf mehrere Objektinstanzen (vom selben oder von unterschiedlichen Objekttypen) aufgerufen werden. Der Zugang zu Daten (d.h. die Leseberechtigungen) wird in Systemen dieser Art anhand von Basisfunktionen für die Einsicht von Objektinstanzen zur Verfügung gestellt. Auf eine Differenzierung zwischen Lese- und Ausführungsrechten kann deshalb im Folgenden verzichtet werden.

5.1 Herausforderung 6: Horizontale und vertikale Bearbeiterzuordnung

In konventionellen WfMS wird jeder interaktiven Aktivität, die eine Aktion des Benutzers erfordert, ein Bearbeiterausdruck zugeordnet (sog. *horizontale Autorisierung*). Dies kann z.B. eine Benutzerrolle oder eine Organisationseinheit aus dem Organisationsmodell sein. Kann eine Aktivität zur Laufzeit aktiviert werden, wird diese allen Benutzern, die sich für diesen Bearbeiterausdruck qualifizieren, innerhalb ihrer Arbeitsliste angeboten. Dies ist innerhalb von datenorientierten Prozess-Management-Systemen nicht mehr ausreichend. Hier müssen, zusätzlich zur Aktivität, die Objektinstanzen, die innerhalb dieser Aktivität verwendet bzw. bearbeitet werden, berücksichtigt werden (sog. *vertikale Bearbeiterzuordnung*) [RoMu97, Mue04].

In einem datenorientierten Prozess-Management-System ist die Auswahl möglicher Bearbeiter nicht nur von der Aktivität abhängig, sondern auch von den Objektinstanzen, die innerhalb einer Aktivität benötigt wird.

Zu jeder Prozessinstanz existiert eine zugehörige Objektinstanz [KuRe09, KuRe09b], deren Attributwerte bei der Ausführung des zugehörigen Prozesses geändert werden. Abbildung 5 illustriert einen Ausschnitt aus verschiedenen Prozessinstanzen für Bewerbungen. Bei einer ausschließlich horizontalen Autorisierung, wie sie in Abbildung 5a zu sehen ist, kann jeder Benutzer, der sich anhand des Bearbeiterausdrucks qualifiziert, die Aktivität `make decision` ausführen. Abbildung 5b verdeutlicht eine gleichzeitige horizontale und vertikale Autorisierung. Ein Benutzer hat die Berechtigung für die Bearbeitung von Bewerbungen derjenigen Bewerber, deren Anfangsbuchstaben zwischen A und L liegen, ein anderer ist für den Bereich zwischen M und Z zuständig.

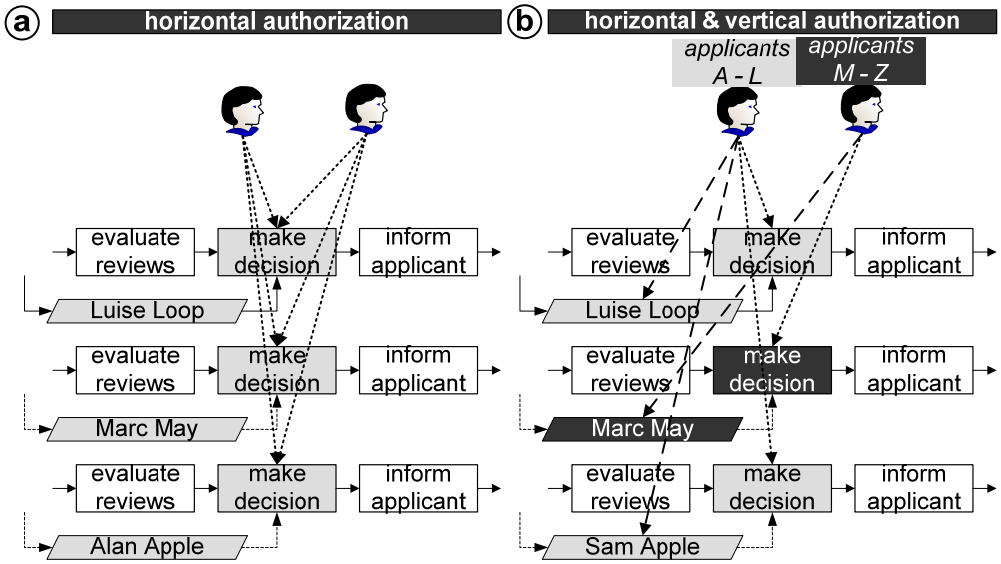

Abbildung 5: Horizontale und vertikale Bearbeiterzuordnung

5.2 Herausforderung 7: Konsistenz zwischen Daten- und Prozessautorisierung

In Datenorientierten Prozess-Management-Systemen werden Prozesse nicht aktivitäts-, sondern datenbezogen modelliert [KuRe09, KuRe09b]. D.h. die Modellierung der Prozessschritte erfolgt nicht wie in konventionellen WfMS auf Basis von Aktivitäten, sondern anhand von Datenbedingungen. Für den Fortschritt einer Prozessinstanz müssen jeweils die innerhalb der Datenbedingung des nachfolgenden Schrittes geforderten Attribute geändert werden. Des Weiteren steht, zusätzlich zur prozessorientierten Sicht (d.h. der Arbeitsliste mit für den Prozessfortschritt obligatorischen Aktivitäten), eine daten- und funktionsbasierte Sicht zur Verfügung. Ausgehend von der daten- und funktionsbezogenen Sicht können optionale Aktivitäten zur Bearbeitung der Attributwerte von Objektinstanzen ausgeführt werden. Optionale Aktivitäten können nicht in die feste Ausführungsreihenfolge der für den Prozess relevanten *obligatorischen Aktivitäten* eingeordnet werden. D.h. die Attributwerte von Objektinstanzen können außerhalb des Prozesses geändert werden. Jedoch müssen bei der Ausführung von optionalen Aktivitäten fehlerhafte Eingriffe in den Prozessverlauf vermieden werden. Optionale Aktivitäten können deshalb nicht vollständig unabhängig von den obligatorischen Aktivitäten der für die Objektinstanz initiierten Prozessinstanz betrachtet werden. Nach der Rückgabe eines `Gutachtens` darf der Mitarbeiter der `Fachabteilung` das Attribut `recommendation` nicht mehr verändern.

In Bezug auf die Autorisierung für optionale Aktivitäten muss der Prozessfortschritt der zur Objektinstanz gehörenden Prozessinstanz berücksichtigt werden.

Für jede Objektinstanz steht zur Laufzeit eine Aktivität zur Verfügung, anhand derer die Attributwerte eingesehen und geändert werden können. Welche Attribute innerhalb dieser Aktivität jeweils zur Verfügung stehen, ist jedoch abhängig vom Fortschritt der zugehörigen Prozessinstanz [Bot02]. Eine Vergabe von beliebigen Berechtigungen kann bei datenorientierten Prozessen zu fehlerhaften Prozessausführungen und -steuerungen führen. Gleichzeitig muss sichergestellt werden, dass für jede obligatorische Aktivität auch ein zuständiger Bearbeiter ermittelt werden kann, d.h. obligatorische Aktivitäten müssen für den Fortschritt des Prozesses zwingend ausgeführt werden.

Für jede obligatorische Aktivität muss mindestens ein Benutzer die Berechtigungen für die Änderung der im Folgeschritt geforderten Attributwerte besitzen.

Abbildung 6 illustriert eine Objekt- und eine Prozessinstanz sowie die einzelnen Berechtigungen für ein Gutachten. Lese- und Schreibberechtigungen werden für die einzelnen Attributwerte in Abhängigkeit vom Fortschritt des Prozesses vergeben. Ein Mitarbeiter der `Fachabteilung` muss zur Bearbeitung eines `Gutachtens` mindestens die Datenberechtigungen zur Änderung der Attribute `recommendation` und `submit` haben. Berechtigungen zur Ausführung der obligatorischen Aktivitäten sind dunkel gekennzeichnet. Dabei muss jedoch berücksichtigt werden, dass z.B. die innerhalb des `Gutachtens` gemachte Rückmeldung, die `recommendation`, nach Rückgabe des Gutachtens an die Personalabteilung nicht mehr verändert werden kann (grau hinterlegte Berechtigungen). Inner-

halb von optionalen Aktivitäten können die Attributwerte einer Objektinstanz jederzeit eingesehen bzw. geändert werden. Hierbei kann gesteuert werden, welche Attribute jeweils zu welchem Zeitpunkt zur Verfügung stehen. Diese Berechtigungen sind in Abbildung 6 weiß dargestellt.

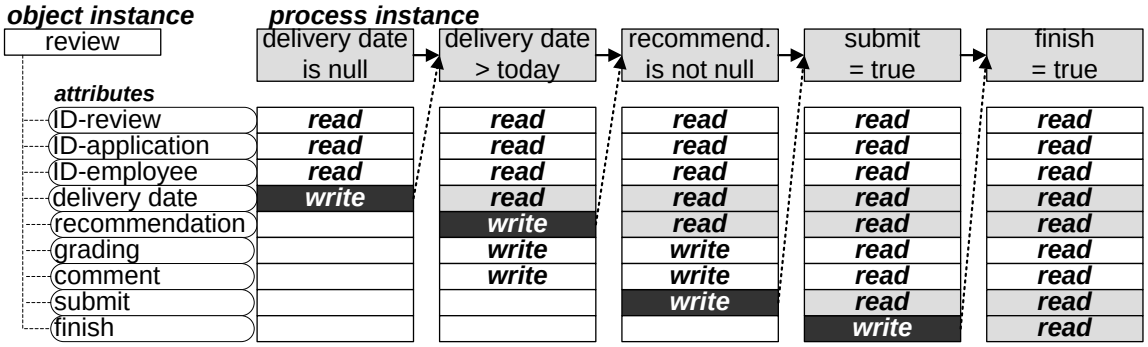

Abbildung 6: Autorisierung für obligatorische Aktivitäten

5.3 Herausforderung 8: Berücksichtigung von Beziehungen zwischen Benutzern und Daten

Im RBAC-Ansatz [SCFY96] kann nicht zwischen Zugriffsrechten in Bezug auf ein individuelles Objekt (d.h. eine Objektinstanz) und Zugriffsrechten in Bezug auf eine (Teil-)Menge von Objektinstanzen eines Objekttyps differenziert werden. Rechte, die ein Benutzer aufgrund seiner Rolle besitzt, müssen auf eine bestimmte Menge von Objektinstanzen beschränkbar sein (siehe Anforderungen in Systemen der Stufe I) [HuWe04, KKC02]. Die Zuordnung zwischen Benutzern und Objektinstanzen ist jedoch in den meisten Situationen nicht willkürlich, sondern unterliegt bestimmten Gegebenheiten [BBU99].

In konventionellen WfMS werden Organisationsmodell und Datenstruktur unabhängig voneinander modelliert und jeweils isoliert voneinander betrachtet. In datenorientierten Prozess-Management-Systemen dagegen sind Benutzer selbst Teil der Datenbasis, d.h. Benutzer sind ebenfalls Objektinstanzen eines bestimmten Objekttyps. Dazu ist eine Integration von Organisationsmodell und Anwendungsdaten, wie sie auch in einigen daten- und funktionsorientierten Anwendungen zu finden ist, nötig. Benutzer und die einzelnen Komponenten der Organisationsstruktur sollten anhand von Objekttypen innerhalb der Datenstruktur modelliert werden können. Benutzer und Organisationseinheiten sind dadurch ebenso Objektinstanzen eines Objekttyps wie alle anderen Anwendungsdaten. Eine Objektinstanz für einen Benutzer kann zur Laufzeit andere Objektinstanzen referenzieren (z.B. für die Zuordnung zu einer Organisationseinheit) und ebenso von anderen Objektinstanzen referenziert werden (z.B. ein Bewerber von einer Bewerbung).

Abbildung 7a verdeutlicht die Integration von Organisationsmodell und Anwendungsdaten. Benutzer sind hier anhand der Objekttypen `Bewerber` und `Mitarbeiter` definiert. Mitarbeiter können Organisationseinheiten, z.B. unterschiedlichen Abteilungen, zugeordnet sein. Anwendungsobjekte sind im dargestellten Beispiel `Ausschreibungen`, `Bewerbungen` und `Gutachten`. Einem Bewerber können dadurch zur Laufzeit direkt mehrere Bewerbungen zugeordnet sein. Jede Ausschreibung ist direkt einem Mitarbeiter aus der Personalabteilung zugeordnet, dieser ist als Sachbearbeiter für die jeweilige Ausschreibung zuständig. Ein Gutachten für eine Bewerbung ist einem Mitarbeiter aus einer Fachabteilung zugeordnet und muss genau von diesem ausgefüllt werden.

Die konkreten Beziehungen zur Laufzeit sind in Abbildung 7b verdeutlicht. Ein Bewerber hat eine Beziehung zu seiner eigenen Bewerbung, nicht aber zu den Bewerbungen von anderen Bewerbern. Ein Mitarbeiter kann sowohl Sachbearbeiter für Ausschreibungen oder Gutachter für eine Menge von Gutachten sein.

Objektinstanzen werden direkt bestimmten Benutzern zugeordnet. Dadurch hat ein Benutzer zur Laufzeit zu verschiedenen Objektinstanzen eines Objekttyps jeweils unterschiedliche Beziehungen.

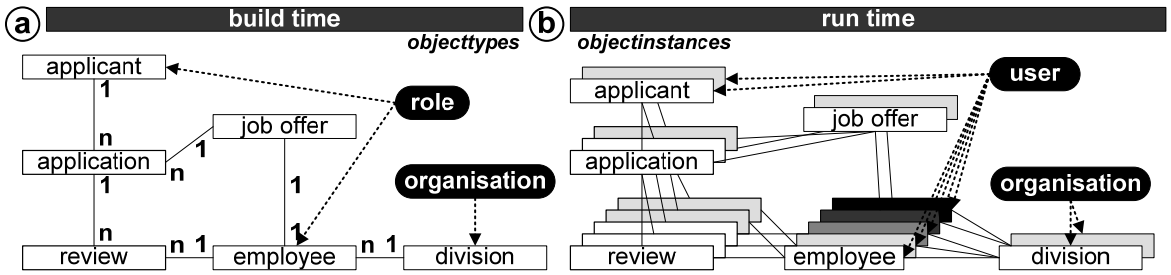

Abbildung 7: Integration von Anwendungsdaten und Organisationsmodell

In konventionellen WfMS werden Berechtigungen bzw. Bearbeiterzuordnungen für Aktivitäten durch Zuordnung von Rollen definiert. Datenorientierte Prozess-Management-Systeme müssen zusätzlich die innerhalb einer Aktivität benötigten Eingabedaten berücksichtigen (siehe Herausforderung 1), d.h. die Objektinstanz die innerhalb der betreffenden Aktivität bearbeitet wird. Die Vergabe von Berechtigungen zur Einsicht und Bearbeitung von Daten erfolgt auf Basis der einzelnen Objekttypen. Hierbei ist jedoch weder die alleinige Zuordnung zu bestimmten Rollen noch die direkte Vergabe an bestimmte Benutzer ausreichend.

Welche Berechtigungen ein Benutzer für eine Objektinstanz besitzt ist nicht nur abhängig von seiner Rolle oder Identität, sondern auch von der jeweiligen Beziehung zwischen dem Benutzer und der betreffenden Objektinstanz [BBU99].

Für die Vergabe von Berechtigungen für einen bestimmten Objekttyp müssen deshalb die einzelnen Rollen durch Berücksichtigung der möglichen Beziehungen weiter verfeinert werden. Die zur Laufzeit möglichen Beziehungen können anhand der Datenrelationen auf Typebene ermittelt werden.

Abbildung 8a zeigt die unterschiedlichen Beziehungen eines Benutzers mit Rolle `Mitarbeiter` in Bezug den Objekttyp `Gutachten`. Für einen Mitarbeiter müssen beispielsweise andere Berechtigungen auf die für ihn angelegten Gutachten im Gegensatz zu den Gutachten von anderen Mitarbeitern vergeben werden. Hierbei sind nicht nur direkte Zuordnungen zwischen Objektinstanzen und Benutzern von Bedeutung.

Neben direkten Zuordnungen zwischen zwei Objektinstanzen sollten indirekte Zuordnungen berücksichtigt werden können.

Indirekte Zuordnungen sind Beziehungen zwischen Objektinstanzen die nicht auf eine unmittelbare Fremdschlüsselbeziehung zurückzuführen sind. Für diese Beziehungen ist eine rekursive Auflösung über mehrere zueinander in Beziehung stehende Objektinstanzen hinweg erforderlich. Dies verdeutlicht Abb. 8. Zu jeder `Bewerbung` existiert eine Menge von `Gutachten`. Ein `Mitarbeiter` hat andere Berechtigungen auf sein eigenes `Gutachten` als auf die `Gutachten` von anderen Mitarbeitern.

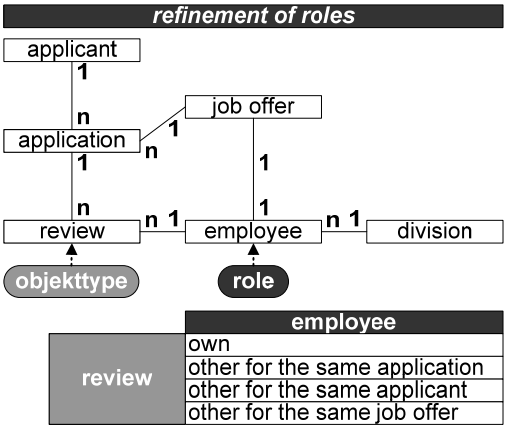

Abbildung 8: Beziehungen zwischen Benutzern und Objektinstanzen

5.4 Herausforderung 9: Differenzierung zwischen Autorisierung und Bearbeiterzuordnung

Hat ein Benutzer Berechtigungen für Daten, die zur Ausführung von obligatorischen Aktivitäten benötigt werden, bedeutet dies nicht zwingend, dass er auch als Bearbeiter für die entsprechenden Aktivitäten ermittelt werden soll.

Bei der Bearbeiterzuordnung für obligatorische Aktivitäten muss differenziert werden, ob ein Benutzer eine Attributänderung durchführen darf oder sie innerhalb des Prozesses zwingend durchführen muss.

Benutzer die eine Änderung zwingend durchführen müssen, sind für die Erfüllung der Datenbedingung eines Prozessschrittes und dadurch für den Fortschritt des Prozesses verantwortlich. Nur für diese Benutzer darf die entsprechende obligatorische Aktivität innerhalb ihrer Arbeitsliste berücksichtigt werden. Andere Benutzer können das Attribut innerhalb einer optionalen Aktivität ändern, sind jedoch nicht zwingend für die Ausführung einer zugehörigen obligatorischen Aktivität verantwortlich. Dadurch sind innerhalb von optionalen Aktivitäten, je nach Vergabe der Datenberechtigungen, auch obligatorische Attributänderungen möglich. Der nächste Fortschritt eines Prozesses kann somit auch ohne die Ausführung einer obligatorischen Aktivität erreicht werden.

Bei einem Wechsel in einen weiterführenden Prozessschritt sollte differenziert werden können, ob dieser Übergang nur explizit, d.h. durch die Ausführung der zugehörigen obligatorischen Aktivität oder auch implizit, d.h. durch die Ausführung einer optionalen Aktivität, erfolgen darf.

In manchen Situationen kann ein impliziter Übergang erlaubt bzw. gewünscht sein. In anderen Fällen ist jedoch eine explizit Überprüfung des angegebenen Attributwerts durch den verantwortlichen Benutzer nötig.

Die beschrieben Herausforderungen verdeutlichen die benötigten Erweiterungen für eine Zugriffskontrolle in datenorientierten Prozess-Management-Systemen. Das Organisationsmodell muss vollständig in die Anwendungsdaten integriert werden. Diese Integration ermöglicht die Modellierung von Beziehungen zwischen Objektinstanzen und Benutzern. Die Bearbeiterzuordnung darf nicht nur *horizontal* (d.h. im Bezug auf Aktivitäten) erfolgen. Zusätzlich müssen die innerhalb der Aktivität verwendeten Eingabedaten berücksichtigt werden. Hier sprechen wir von einer sog. *vertikalen Autorisierung*. Parallel zu den für die Ausführung der Prozesse entscheidenden obligatorischen Aktivitäten können optionale Aktivitäten ausgeführt werden. Die Autorisierung erfolgt in Bezug auf die einzelnen Objekttypen. Hierbei müssen jedoch die verschiedenen Schritte des zugehörigen Prozesstyps berücksichtigt werden. Sowohl bei der Autorisierung zur Datenbearbeitung als auch bei der Bearbeiterzuordnung müssen die unterschiedlichen Beziehungen, die zwischen Benutzern und Objektinstanzen möglich sind, berücksichtigt werden. Dazu ist eine Verfeinerung der Rollen nötig. Bei der Autorisierung muss zwischen obligatorischen und optionalen Rechten unterschieden werden. Benutzer mit obligatorischen Rechten sind für die Ausführung prozessrelevanter Attributänderungen verantwortlich.

6 Verwandte Arbeiten

Einige der beschriebenen Herausforderungen wurden im Ansatz bereits von anderen Autoren adressiert. Diese stellen jedoch nur Teillösungen für einzelne Problematiken bereit.

6.1 Herausforderung 6: Horizontale und vertikale Bearbeiterzuordnung

In speziellen, auf WfMS ausgerichteten Ansätzen hat die Bearbeiterzuordnung immer eine höhere Priorität als die benötigten Datenberechtigungen. [ThSa97] beschreibt einen allgemeinen Ansatz, in welchem einzelne Rechte innerhalb eines sog. Autorisierungsschritts gekapselt werden. Ein Recht beschreibt jeweils eine mögliche Aktion eines Benutzers in Bezug auf einen Objekttyp im Kontext einer bestimmten Aufgabe. Zur Laufzeit werden die Rechte jedoch nicht automatisch vom System aktiviert. Die Vergabe erfolgt explizit durch Zustimmung anderer Benutzer zu dem Zeitpunkt, an dem ein Benutzer die Berechtigung anfordert. Dadurch kann die Berechtigung zur Ausführung der jeweiligen Aufgabe, unter Berücksichtigung der betroffenen Objektinstanz, aktiviert werden, obwohl die Rechte auf der Basis von Objekttypen definiert werden.

[SSML02] dagegen ist ein speziell auf WfMS ausgerichteter Ansatz. Dieser beschreibt das Konzept der sog. "instanzbezogenen Benutzergruppe". Jeder Benutzer, der Bearbeiter von mindestens einer

Aktivität einer Prozessinstanz war, bekommt (entsprechend seiner Rolle) Zugriff auf die innerhalb der Prozessinstanz verwendeten Objektinstanzen. Dadurch wird ein Bezug zwischen Prozessinstanzen einerseits und Objektinstanzen andererseits berücksichtigt. Allerdings hat auch hier die Bearbeiterzuordnung eine höhere Priorität als die Zugriffsrechte auf Daten. Erst nach Zuordnung eines Benutzers als Bearbeiter werden die entsprechenden Datenzugriffsrechte aktiviert. Ein Benutzer sollte jedoch nur dann als Bearbeiter einer Aktivität überhaupt ausgewählt werden können, wenn er auch die zugehörigen Berechtigungen für die benötigten Objektinstanzen besitzt.

Diese Anforderung wird auch in [RoMu97] diskutiert. Dieser Ansatz beschreibt eine sog. Prozessobjekthierarchie. Für jedes Objekt werden innerhalb des WfMS die prozessrelevanten Eigenschaften verwaltet und die jeweils zuständigen Rollen verwaltet. Bei der Aktivität werden zusätzlich zum Bearbeiterausdruck zusätzlich die jeweils relevanten Objekteigenschaften angegeben. Dies führt jedoch zu einer redundanten Datenhaltung, da die für die Autorisierung relevanten Daten sowohl im WfMS als auch in den externen Anwendungen verwaltet werden. Dadurch entstehen Inkonsistenzen und ein erhöhter Pflege- und Verwaltungsaufwand.

6.2 Herausforderung 7: Konsistenz zwischen Daten- und Prozessautorisierung

Alle Ansätze in diesem Bereich beziehen sich auf die Vergabe von Datenberechtigungen bei der Ausführung einer zu einem Prozess gehörenden Aktivität. Die Bearbeitung von Daten außerhalb der Aktivitäten eines Prozesses wird bisher in keinem Ansatz berücksichtigt. Für die Ausführung einer Aktivität kann grob zwischen Ansätzen mit impliziten und Ansätzen mit expliziten Datenberechtigungen differenziert werden. Implizite Datenberechtigungen werden anstatt Benutzern oder Rollen direkt den jeweiligen Aktivitäten zugeordnet. Benutzer, die zur Laufzeit die entsprechende Aktivität bearbeiten, bekommen automatisch die zugehörigen Datenberechtigungen zugeordnet. Diese sind jeweils für den Zeitraum der Ausführung der Aktivität gültig. In [AWG05] wird zwischen obligatorischen und optionalen Datenelementen innerhalb einer Aktivität differenziert. Es werden jedoch nur Ausführungsrechte für Aktivitäten und keine expliziten Berechtigungen für die einzelnen Datenelemente definiert (d.h. implizite Rechte für Datenzugriffe). Jeder Benutzer, der als Bearbeiter für eine Aktivität innerhalb eines sog. "Case" (d.h. einer Prozessinstanz) zuständig ist, erhält auch Zugang zu allen Datenelementen, die innerhalb dieses "Case" zur Verfügung stehen. Ein ähnlicher Ansatz ist in [SSML02] beschrieben. Ein Benutzer erhält hier bei der Ausführung einer Aktivität Zugang zu allen Daten, die innerhalb einer Prozessinstanz, in der er als Bearbeiter beteiligt ist oder war, verwendet wurden.

[Bot02] geht über diese Ansätze hinaus. Hier können einzelne Berechtigungen für Datenzugriffe auf Attributebene pro Aktivität, also in Abhängigkeit vom Prozessfortschritt, jeweils unterschiedlich definiert werden. Verschiedene Objektinstanzen eines jeweiligen Objekttyps können hierbei allerdings ebenso nicht berücksichtigt werden. Bei expliziten Datenberechtigungen wie sie beispielsweise in [ThSa97] beschrieben werden, werden Datenberechtigungen weiterhin Benutzern zugeordnet. Zusätzlich kann jedoch der jeweilige Ausführungskontext berücksichtigt werden. In [ThSa97] werden hierzu einzelne Berechtigungen zu einem Ausführungsschritt zusammen und damit zusammengehörige Rechte im Bezug auf einzelne Aufgaben beschrieben.

6.3 Herausforderung 8: Berücksichtigung von Beziehungen zwischen Benutzern und Daten

In der Literatur existieren verschiedene Ansätze, die die Einschränkung der Rechte einer Rolle auf eine bestimmte Menge von Objektinstanzen erlauben [LuSl97, KKC02, T97, HW04, BBU99]. Nicht alle erlauben jedoch eine Einschränkung der Objektinstanzen in Abhängigkeit von den jeweiligen Beziehungen zu Benutzer.

In [KKC02] können dazu pro Benutzer und Objekt verschiedene sicherheitsrelevante Informationen definiert werden. Innerhalb der Rollen können Bedingungen in Bezug auf Benutzer- und Objektinformationen definiert werden. Dadurch können verschiedene Beziehungen zwischen Benutzern und Objektinstanzen beschrieben werden. Dieser Ansatz geht jedoch von einer getrennten Datenhaltung des Organisationsmodells und der Anwendungsdaten aus, und erfordert deshalb einen sehr hohen manuellen Administrationsaufwand. Informationen und Beziehungen, die innerhalb einer integrierten Sicht auf Anwendungsdaten und Organisationsmodell bereits vorhanden sind, müssen manuell definiert werden. Auch [Tho97] beschreibt einen Benutzer- und einen Objektkontext. Diese sind jeweils innerhalb eines sog. Teams gültig. Neben der Gruppierung von Benutzern zu Rollen können diese zusätzlich in Teams zusammengefasst werden. Benutzer desselben Teams haben jeweils Zugriff auf eine bestimmte Menge an Objektinstanzen. Dieser Ansatz ist jedoch weniger flexibel da die Beziehungen

zwischen Benutzern und Objekten zur Modellierzeit definiert werden müssen. Ein ähnliches Vorgehen wird in [HuWe04] beschrieben. Hier können Beziehungen zwischen Subjekten und Objekten anhand eines so genannten "Context Type" beschrieben werden. Ein Kontexttyp ist zunächst ein einfaches Merkmal eines Benutzers, z.B. behandelnder Arzt. In Bezug auf die Kontexttypen können Bedingungen definiert werden, die jeweils den einzelnen Rechten zugeordnet werden. [BBU99] geht konkret auf Beziehungen zwischen Benutzern und Objektinstanzen ein. Diese können trotz einer Trennung von Anwendungsdaten und Organisationsmodell berücksichtigt werden. Indirekte Beziehungen, die eine rekursive Auflösung erfordern, können jedoch in keinem der Ansätze ausgedrückt werden.

6.4 Herausforderung 9: Differenzierung zwischen Autorisierung und Bearbeiterzuordnung

Ansätze, die zumindest teilweise auf Anforderungen dieser Art eingehen, können grob in zwei Gruppen eingeteilt werden. In die erste Gruppe können Ansätze eingeordnet werden, die auf verschiedene Arten von Bearbeiterzuordnungen eingehen [RHE04, BFA99, WBK03]. Der in [AWG05] vorgestellte Ansatz dagegen kann einer anderen Gruppe zugeordnet werden. Er differenziert in Bezug auf die Datenzugriffsrechte innerhalb einer Aktivität zwischen optionalen und obligatorischen Datenelementen.

In [RHE04] werden verschiedene Zuordnungs- und Aktivierungsmöglichkeiten von Aktivitäten für Benutzer beschrieben. Dabei wird unter anderem zwischen einer direkten Zuordnung einer Aktivität vom System und zwischen einer freiwilligen Auswahl vom Benutzer unterschieden. In [BFA99] werden Rollen hierarchisch angeordnet. Die Zuordnung einer Aktivität erfolgt bei mehreren in Frage kommenden Benutzern jeweils an den mit der hierarchisch tiefer angeordneten Rolle. Des Weiteren kann direkt zwischen optionalen und obligatorischen Bearbeiterzuordnungen differenziert werden. In [WBK03] dagegen können Prioritäten innerhalb der Bearbeiterzuordnungen angegeben werden.

7 Vision und Ausblick

Unser Ziel ist die Entwicklung eines umfassenden und nachhaltigen Rahmenwerks für ein datenorientiertes Prozess-Management-System. Dadurch soll die generische Unterstützung von datenorientierten Prozessen mit einer engen Integration von Daten, Prozessen und Benutzer möglich werden. Das System soll die Funktionalität von datenorientierten Anwendungen abbilden können und gleichzeitig die Vorteile bieten, die konventionelle Workflow-Management-Systeme mit sich bringen. In [KuRe09, KuRe09b] haben wir dazu fünf fundamentale Herausforderungen in Bezug auf die Integration von Daten und Prozessen definiert. Diese haben wir in diesem Aufsatz um vier weitere grundlegende Herausforderungen im Hinblick auf die Integration von Benutzern ergänzt.

Aktuell entwickeln wir eine grundlegend neue Architektur für ein datenorientiertes Prozess-Management-System. Innerhalb dieses Rahmenwerks sollen die beschriebenen Anforderungen umgesetzt werden können. Weitere Arbeiten bilden die detaillierte Beschreibung der einzelnen Komponenten und deren Zusammenhänge.

Literatur

[AaHe04] W. M. P. van der Aalst, K. van Hee: *Workflow-Management - Models, Methods and Systems*. MIT Press, 2004.

[AHKB03] W.M.P van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, A.P. Barros: *Workflow Patterns*, Distributed and Parallel Databases, 14(3):5-51, 2003.

[AWG05] W.M.P. van der Aalst, M. Weske, D. Grünbauer: *Case Handling: A New Paradigm for Business Process Support.* Data & Knowledge Engineering, 53:129-162, 2005.

[BBR06] R. Bobrik, T. Bauer, M. Reichert: *Proviado – Personalized and Configurable Visualizations of Business Processes*. In: Proc. 7th Int'l Conf. on Electronic Commerce and Web Technologies (EC-WEB'06), Krakow, Poland, Springer, LNCS 4082, pp. 61-71, 2006

[BBU99] J. Barkley, K. Beznosov, J. Uppal: *Supporting Relationships in Access Control Using Role-based Access Control*, RBAC '99: Proc. RBAC'99, pp. 55-65, 1999

[Ber98] E. Bertino: *Data Security.* Data Knowledge Engineering, 25(1-2):199-216, 1998.

[BFA99] E. Bertino, E. Ferrari, V. Atluri: *The Specification and Enforcement of Authorization Constraints in Workflow Management Systems,* ACM Transactions on Information and System Security, 2(1): 65-104, 1999

[Bot02] R. A. Botha: *CoSAWoE – A Model for Context-sensitive Access Control in Workflow Environments,* PhD thesis, Rand Afrikaans University, 2002

[BRBB09] S. Bassil, M. Reichert, R. Bobrik, T. Bauer: *Access Control for Monitoring System-Spanning Business Processes in Proviado.* In: Proceedings 3rd Int'l Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'09), Lecture Notes in Informatics (LNI) P-152, pp. 125-139, 2009

[DaRe09a] P. Dadam, M. Reichert: *The ADEPT Project: A Decade of Research and Development for Robust and Flexible Process Support - Challenges and Achievements*. Computer Science - Research and Development, 23(2): 81-97, 2009

[DaRe09b] P. Dadam, M. Reichert, S. Rinderle-Ma, K. Goeser, U. Kreher, M. Jurisch: *Von ADEPT zur AristaFlow BPM Suite - Eine Vision wird Realität: "Correctness by Construction" und flexible, robuste Ausführung von Unternehmensprozessen*. EMISA Forum, 29(1): 9-28, 2009.

[FeKu92] D. Ferraiolo, R. Kuhn: *Role-based Access Control,* 15th National Computer Security Conference, 1992

[HuWe04] J. Hu, A. C. Weaver: *A Dynamic, Context-Aware Security Infrastructure for Distributed Healthcare Applications*, Pervasive Security, Privacy and Trust (PSPT2004), Boston, MA, August 2004.

[Kan99] M. Kang, J. Froscher, A. Sheth, K. Kochut, J. Miller*: A Multilevel Secure Workflow Management System,* Proc. CAiSE 1999, pp. 271 – 285

[KaSa01] S. Kandala, R. Sandhu: *Secure Role-Based Workflow Models,* Proc. 5th Conference on Database and Application Security, pp. 45 - 58, 2001

[Kin97] T. Kindler: *ABAC: Activity-based Security in Intranet and Internet Workflows,* Proc. WITS, 1997

[KKC02] A. Kumar, N. Karnik, G. Chafle: *Context Sensitivity in Role-based Access Control,* ACM SIGOPS Operating Systems Review, 36: 53 - 66, 2002

[KuRe09] V. Künzle, M. Reichert: *Towards Object-aware Process Management Systems: Issues, Challenges, Benefits*, In: Proceedings 10th Int'l Workshop on Business Process Modeling, Development, and Support (BPMDS'09), Amsterdam, Springer, LNBIP 29, pp. 197-210, 2008

[KuRe09b] V. Künzle, M. Reichert: *Herausforderungen auf dem Weg zu datenorientierten Prozess-Management-Systemen.* EMISA Forum, 29(2): 9-24, 2009.

[KuRe09c] V. Künzle, M. Reichert: *Integrating Users in Object-aware Process Management Systems: Issues and Challenges.* Proceedings 5th International Workshop on Business Process Design (BPD09), Ulm, 2009

[LRDR06] L.T. Ly, S. Rinderle, P. Dadam, M. Reichert: *Mining Staff Assignment Rules from Event-Based Data*. In: Proceedings BPM'05 workshops. Nancy, Springer, LNCS 3812, pp. 177-190, 2006

[LuSl97] E. Lupu, M. Sloman: *A Policy Based Role Object Model*, Proc. EDOC'97, pp. 36-47, 1997

[MRH08] D. Müller, M. Reichert, J. Herbst: *A New Paradigm for the Enactment and Dynamic Adaptation of Data-driven Process Structures*. In: 20th Int'l Conf. on Advanced Information Systems Engineering (CAiSE'08), Montpellier, France. Springer, LNCS 5074, pp. 48-63, 2008

[MRH07] D. Müller, M. Reichert, J. Herbst: *Data-driven Modeling and Coordination of Large Process Structures*. In: Proceedings of the 15th Int'l Conf. on Cooperative Information Systems (CoopIS'07), Vilamoura, Portugal. Springer, LNCS 4803, pp. 131-149

[Mue04] M. zur Mühlen: *Organizational Management in Workflow Applications: Issues and Perspectives*, Information Technology and Management, 5(3-4): 271-291, 2004

[OSM00] S. Osborn, R. Sandhu, Q. Munawer: *Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies,* ACM TISSEC, 3: 85 - 106, 2000

[Pfe05] V. Pfeiffer: *A Framework for Evaluating Access Control Concepts in Workflow Management Systems*, Master thesis, University of Ulm, 2005 (in German)

[ReDa00] M. Reichert, P. Dadam: *Geschäftsprozessmodellierung und Workflow-Management - Konzepte, Systeme und deren Anwendung.* Industrie Management, 16(3): 23-27, 2000.

[Rei00] M. Reichert: *Dynamische Ablaufänderungen in Workflow-Management-Systemen.* Dissertation, Universität Ulm, 2000

[RHE04] N. Russell, A. ter Hofstede, D. Edmond: *Workflow Resource Patterns*, Proc. CAISE'05, 2004

[ReRD09] M. Reichert, S. Rinderle-Ma, P. Dadam: *Flexibility in Process-Aware Information Systems*. In: Transactions on Petri Nets and other Models of Concurrency 2: 115-135, 2009

[RiRe05] S. Rinderle, M. Reichert: *On the Controlled Evolution of Access Rules in Cooperative Information Systems*. In: Proc. 13th Int'l Conf. on Cooperative Information Systems (CoopIS'05), Agia Napa, Cyprus. Springer, LNCS 3760, pp. 238-255, 2005

[RiRe06] S. Rinderle, M. Reichert: *Data-Driven Process Control and Exception Handling in Process Management Systems*. In: Proc. 18th Int'l Conf. on Advanced Information Systems Engineering (CAiSE'06), Luxembourg, Springer, LNCS 4001, pp. 273-287, 2006

[RiRe07] S. Rinderle-Ma, M. Reichert: *A Formal Framework for Adaptive Access Control Models.* Springer, Journal on Data Semantics IX , Vol. LNCS 4601, 2007, pp. 82-112

[RiRe08] S. Rinderle-Ma, M. Reichert: *Managing the Life Cycle of Access Rules in CEOSIS*. In: Proceedings of the 12th IEEE International Enterprise Computing Conference (EDOC'08), September, 2008, Munich, Germany, pp. 257-266

[RiRe09] S. Rinderle-Ma, M. Reichert: *Comprehensive Life Cycle Support for Access Rules in Information Systems: The CEOSIS Project.* Enterprise Information Systems, 3(3): 219-251, 2009

[RoMu97] M. Rosemann, M. zur Mühlen: *Modellierung der Aufbauorganisation in Workflow-Management-Systemen: Kritische Bestandsaufnahme und Gestaltungsvorschläge*, EMISA-Forum, 3(1):78-86, 1998

[San00] R. Sandhu: *Engineering Authority and Trust in Cyberspace: The OM-AM and RBAC Way,* Proc. 5th ACM workshop on Role-based Access Control, pp. 111 - 119, 2000

[SaVi01] P. Samarati, S. Vimercati: *Access Control: Policies, Models, and Mechanisms,* Foundations of Security Analysis and Design, 2001

[SCFY96] R. Sandhu, E. Coynek, H. Feinsteink, C. Youmank: *Role-Based Access Control Models,* IEEE Computer, 29: 38-47, 1996

[SSML02] S. Wu, A. Sheth, J. Miller, Z. Luo: *Authorization and Access Control Of Application Data In Workflow-Systems,* Journal of Intelligent Information Systems, 18: 71 - 94, 2002

[Tho97] R. Thomas: *Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments*, Proc. RBAC'97, pp.13-19, 1997

[ThSa97] R. K. Thomas, R. S. Sandhu: *Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management,* Proc. IFIP TC11 WG11.3 Workshop on Database Security, 113: 166 -181, 1997

[WBK03] J. Wainer, P. Barthelmess, A. Kumar: *W-RBAC – A Workflow Security Model incorporating controlled overriding of Constraints***,** Int. Journal on Cooperative Information Systems, 12(4): 455-485, 2003

[WSR09] B. Weber, S. Sadiq, M. Reichert: *Beyond Rigidity - Dynamic Process Lifecycle Support: A Survey on Dynamic Changes in Process-aware Information Systems.* Computer Science - Research and Development, 23(2):47-65, 2009

[WRWR05] B. Weber, M. Reichert, W. Wild, S. Rinderle: *Balancing Flexibility and Security in Adaptive Process Management Systems.* In: Proc. 13th Int'l Conf. on Cooperative Information Systems (CoopIS '05), Agia Napa, Cyprus. Springer, LNCS 3760, pp. 59-76, 2005